IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Jean-Yves SIMON, et al.	§	Confirmation No.:	9476
		§		
Serial No.:	10/764,670	§	Group Art Unit:	2112
		§		
Filed:	January 26, 2004	§	Examiner:	Fritz Alphonse
		§		
For:	Method for Streamlining Error	§	Docket No.:	TI-36989
	Correction Code Computation	§		(1962-09800)
	While Reading or	§		
	Programming a NAND	§		
	Flash Memory	§		

APPEAL BRIEF

Mail Stop Appeal Brief – Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Date: October 29, 2008

Sir:

Appellants hereby submit this Appeal Brief in connection with the above-identified application. A Notice of Appeal is filed concurrently herewith.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is Texas Instruments Inc., a Delaware corporation, having its principal place of business in Dallas, Texas. The Assignment from the inventors to Texas Instruments was recorded on January 26, 2004, at Reel/Frame 014930/0563.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF THE CLAIMS

Originally filed claims:	1-26.
Claim cancellations:	None.
Added claims:	None.
Presently pending claims:	1-26.
Presently appealed claims:	1-26.

IV. STATUS OF THE AMENDMENTS

No claims were amended after the final Office action dated September 2, 2008.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The specification is directed to a method for streamlining Error Correction Code (ECC) computation while reading or programming a NAND flash memory. **{Specification Title}**.[1] At least some of the illustrative embodiments are methods as in claim 1:

1. A method, comprising:
transferring a data block between a flash memory and a memory controller; **{12, [0033], lines 8-10, Figure 4a, element 408}**
computing an ECC for said data block while transferring the data block; and **{12, [0033], lines 8-10, Figure 4a, element 408}**
selectively storing the ECC in a plurality of registers using a switching mechanism, the storing while transferring the data block. **{12, [0033], lines 8-10, Figure 4a, element 408}**

Other embodiments are systems as in claim 6:

6. A system, comprising:
a flash memory; **{3, [0016], lines 1-2, Figure 1a, element 114}**
a controller coupled to the flash memory; **{3, [0016], lines 1-2, Figure 1a, element 100}**
a switch coupled to the controller; and **{10, [0027], lines 2-4, Figure 1b, element 238}**
said controller is configured to shift a data block between the flash memory and the controller, while computing an ECC for said data block; and **{11, [0030], lines 1-7, Figure 2, element 200}**
said system is configured to selectively store the ECC in a plurality of registers using the switch, while the controller shifts the data block. **{11. [0030], lines 1-7, Figure 2, element 238}**

Yet still other embodiments are systems as in claim 12:

12. A system comprising:
a means for storing[2] a data block; **{3, [0016], lines 1-2, Figure 1a, element 114}**
a means for controlling[3] the data block; **{3, [0016], lines 1-2, Figure 1a, element 100}**

[1] For consistency, citations to the Specification will take the form **{[page], [paragraph number], lines [lines within the paragraph]}.**

[2] This limitation is specifically identified as a Means plus Function under 35 U.S.C § 112, 6th.

[3] This limitation is specifically identified as a Means plus Function under 35 U.S.C § 112, 6th.

a means for computing[4] an ECC of the data block; **{11, [0030], lines 3-6, Figure 2, element 202}**
a means for shifting[5] the data block between the means for storing and the means for controlling while computing an ECC for said data block; and **{11, [0030], lines 1-7, Figure 2, element 200}**
a means for selectively storing[6] the ECC in a plurality of registers while shifting the data block. **{11, [0030], lines 1-7, Figure 2, element 238}**

Other embodiments are memory controllers as in claim 18:

18. A memory controller configured to couple to a memory, comprising:
a memory interface; **{10, [0027], lines 5-8, Figure 2, element 204}**
an ECC engine configured to compute an ECC while transferring a data block between the ECC engine and memory; and **{11, [0030], lines 3-6, Figure 2, element 202}**
a switching mechanism coupled to the ECC engine, the ECC engine configured to selectively store the ECC in a plurality of registers using the switching mechanism, while transferring the data block. **{11, [0030], lines 1-7, Figure 2, element 238}**

[4] This limitation is specifically identified as a Means plus Function under 35 U.S.C § 112, 6th.

[5] This limitation is specifically identified as a Means plus Function under 35 U.S.C § 112, 6th.

[6] This limitation is specifically identified as a Means plus Function under 35 U.S.C § 112, 6th.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claim 1 is indefinite under 35 U.S.C. §112, second paragraph.

Whether claims 1-4, 6-10, 12-16 and 18-26 are obvious under 35 U.S.C. §103(a) over Eggleston et al. (US Pat. No. 9,906,961, hereinafter 'Eggleston'), Wei et al. (US Pat. No. 6,683,817, hereinafter 'Wie') and Ito et al. (US PGPub No. 2004/0221098, hereinafter 'Ito').[7]

Whether claims 5, 11 and 17 are obvious under 35 U.S.C. §103(a) over Eggleston, Wei, Ito and Acton (US Pat. No. 6,883,131, hereinafter 'Acton').

[7] In the body of the rejections it appears that claim 12 stands rejected over Eggleston, Kikuchi et al. (US Pat. No. 6,131,139, hereinafter 'Kikuchi') and Ito.

VII. ARGUMENT

A. Section 112 Rejections

Claim 1 stands rejected as allegedly indefinite under 35 U.S.C. 112, second paragraph. In particular, the Office Action dated September 2, 2008 states that "As to claim 1, it is not clear to what is meant by the limitation "the storing while transferring the data block." Appellants respectfully traverse. Paragraph [0033] of the specification recites, "The starting address of the NAND Flash memory 214 where data may be written also may be sent to the NAND Flash memory 214 (block 406). Data then may be sent to the NAND Flash memory 214 and the ECC Engine 202 may concurrently compute the ECC and store the ECC in the current ECC register 218-234 (block 408)." Accordingly, Appellants submit that the ECC is calculated and stored while the transferring data to the flash memory.

Based on the foregoing, Appellants respectfully request that the 35 U.S.C. 112, second paragraph rejection be reversed, and the claim set for issue.

B. Section 103 Rejections over Eggleston, Wei and Ito

1. Claims 1-4, 6-10 and 18-26

Claims 1-4, 6-10 and 18-26 stand rejected as allegedly obvious over Eggleston, Wei and Ito. Claim 1 is representative of this grouping of claims. The grouping should not be construed to mean the patentability of any of the claims may be determined in later actions (*e.g.*, actions before a court) based on the groupings. Rather, the presumption of 35 USC § 282 shall apply to each of these claims individually.

Ito is directed to a semiconductor integrated device.[8] Ito teaches an ECC-CODEC that produces check bits for error correction/detection with reference to data in the memory, and stores the check bits in a pre-determined region of the memory.[9] When the ECC-CODEC is operated as a coder the data read from the memory is supplied to feedback shift registers to calculate check bits based on the data read from the memory.

> In case where the ECC-CODEC circuit 7 is operated as the coder circuit, readout data (i.e., original memory data) are sent from the main

[8] Ito Title.

[9] Ito Abstract.

> amplifier MA (or the data output register storing the readout data form the memory) through an AND gate 71 controlled by the SYNDROME signal and an EX-OR (Exclusive OR) circuit 72 to be supplied to a circulating circuit 73 comprising a plurality of feedback shift registers (FSR) S0 to S15 of a left/right shit type and EX-OR circuits. After subjected to logical operation, the readout data are sent through a switch 74 controlled by a parity (PARITY) signal to be delivered as the parity data to the write buffer WB (or the data input register) and written into the memory or a parity data accumulating section as write data.[10]

Moreover, Ito teaches that the parity data is delivered to write buffer in the cycle following the calculation.

> In addition, the read data are taken into the shift registers (S0-S15) of the circulating circuit 73 in FIG. 6 and subjected to calculation to produce the parity data based on the original memory data. In a following cycle, the parity data are delivered to the write buffer WB bit by bit. … Simultaneously, the row address (XA) and the column address (YA) corresponding to the parity bit region are acquired. Based on these address, Column (column address) is incremented and 16 parity bits are written into memory cells … .[11]

Thus, Ito appears to teach loading data read from the memory into feedback shift registers that are used to perform the logical operation to calculate check bits, and storing the calculated check bits into the memory or a parity data accumulating section. Ito also teaches that the ECC-CODEC calculates check bits for data already in memory.

Representative claim 1, by contrast, specifically recites “computing an ECC for said data block while transferring the data block; and selectively storing the ECC in a plurality of registers using a switching mechanism, the storing while transferring the data block.” Appellants submit that Eggleston, Wei and Ito do not teach or fairly suggest such a method. Ito appears to teach loading data read from the memory into feedback shift registers that are used to perform the logical operation to calculate check bits, and storing the calculated check bits into the memory or a parity data accumulating section. Ito does not appear to teach selectively storing the check bits in a plurality of registers. Ito also teaches that the ECC-CODEC calculates check bits for data already in memory. Thus,

[10] Ito Paragraph [0143].

[11] Ito Paragraph [0144].

even if the teachings of Eggleston and Wei are precisely as the Office Action suggests (which Appellants do not admit), Eggleston, Wei and Ito still fail to teach or fairly suggest "computing an ECC for said data block **while transferring the data block**; and **selectively storing the ECC in a plurality of registers** using a switching mechanism, **the storing while transferring the data block**."

Moreover, in the *Response to Arguments* section of the final Office Action dated September 2, 2008, the Office Action states that Eggleston discloses a system

> [T]hat stores a first portion of the ECC in a first register; and storing a second portion of the ECC in an alternate register if the first register is full (fig. 8; col. 16, lines 45 through col. 17 line 5). Eggleston's system includes a controller which transfers contents of the registers to the memory (col. 16, lines 9-30).[12]

Appellants respectfully traverse. Eggleston is directed towards erase block data splitting.[13] In particular, Eggleston teaches flash memory devices that split the user data from the associated overheard data among separate flash memory devices to avoid the issue of potential corruption.[14] In col. 16, lines 45 through col. 17, line 5, Eggleston teaches that ECC hardware generates the required ECC for the user data written into or read from the logic sectors of the write/read accessed physical sector and temporarily stores the ECC in a RAM storage circuit. When the next physical sector is write/read accessed, the ECC from the RAM storage circuit is written out to the ECC code area of the next physical sector. Thus, Appellants submits that Eggleston teaches temporarily storing ECC in RAM storage before the ECC is written to the ECC code area of a physical sector. Eggleston does not teach or fairly suggest "computing an ECC for said data block while transferring the data block; and selectively storing the ECC in a plurality of registers using a switching mechanism, the storing while transferring the data block."

Based on the foregoing, Appellants respectfully request that the rejection of this grouping be reversed, and the claims set for issue.

[12] Final Office Action date September 2, 2008, page 6.

[13] Eggleston Title.

[14] Eggleston Col. 2, lines 60-67.

2. Claims 12-16

Claims 12-16 stand rejected as allegedly obvious over Eggleston, Wei and Ito. Claim 12 is representative of this grouping of claims. The grouping should not be construed to mean the patentability of any of the claims may be determined in later actions (*e.g.*, actions before a court) based on the groupings. Rather, the presumption of 35 USC § 282 shall apply to each of these claims individually.

In the *Response to Arguments* section of final Office Action dated September 2, 2008, the Office Action by law states

> Eggleston does not explicitly teach computing an ECC of the data block; storing the ECC in a plurality of registers.[15]

Thus, based on the foregoing, Appellants respectfully request that the rejections of this grouping be reversed, the claims set for issue.

Moreover, In the *Response to Arguments* section of final Office Action dated September 2, 2008, the Office Action by law states “Eggleston does not explicitly teach computing an ECC of the data block; storing the ECC in a plurality of registers. However, the limitations are disclosed by Kikuchi (figs. 2, 11, col. 16, lines 30-41).[16] Appellants respectfully traverse. Kikuchi is directed to apparatus and method of simultaneously reading and writing data in a semiconductor device having a plurality of flash memories.[17] In particular, Kikuchi teaches a flash disk card mounted with a controller and a plurality of flash memories, and the flash disk card is connected to a card slot of a host computer.[18] Kikuchi also teaches an error controller to perform an ECC process when writing/reading is performed.[19] In col. 6, lines 30-41, Kikuchi teaches shifting, one page of data at time, the data in a data block in one of the plurality of flash memories to an empty data block in another one of the plurality of memories. Appellants respectfully submits that Kikuchi appears to teach shifting data between the plurality of flash memories connected to the

[15] Final Office Action dated September 2, 2008, page 7

[16] Final Office Action dated September 2, 2008, page 7

[17] Kikuchi Title.

[18] Kikuchi Col. 5, lines 32-40.

host computer, but Kikuchi is silent as to storing ECC in a plurality of registers while shifting the data block. Thus, Appellants submit that Kikuchi does not teach "a means for shifting the data block between the means for storing and the means for controlling while computing an ECC for said data block; and a means for selectively storing the ECC in a plurality of registers while shifting the data block."

Additionally, Ito is directed to a semiconductor integrated device.[20] Ito teaches an ECC-CODEC that produces check bits for error correction/detection with reference to data in the memory, and stores the check bits in a pre-determined region of the memory.[21] When the ECC-CODEC is operated as a coder the data read from the memory is supplied to feedback shift registers to calculate check bits based on the data read from the memory.

> In case where the ECC-CODEC circuit 7 is operated as the coder circuit, readout data (i.e., original memory data) are sent from the main amplifier MA (or the data output register storing the readout data form the memory) through an AND gate 71 controlled by the SYNDROME signal and an EX-OR (Exclusive OR) circuit 72 to be supplied to a circulating circuit 73 comprising a plurality of feedback shift registers (FSR) S0 to S15 of a left/right shit type and EX-OR circuits. After subjected to logical operation, the readout data are sent through a switch 74 controlled by a parity (PARITY) signal to be delivered as the parity data to the write buffer WB (or the data input register) and written into the memory or a parity data accumulating section as write data.[22]

Moreover, Ito teaches that the parity data is delivered to write buffer in the cycle following the calculation.

> In addition, the read data are taken into the shift registers (S0-S15) of the circulating circuit 73 in FIG. 6 and subjected to calculation to produce the parity data based on the original memory data. In a following cycle, the parity data are delivered to the write buffer WB bit by bit. … Simultaneously, the row address (XA) and the column address (YA) corresponding to the parity

[19] Kikuchi Col. 6, lines 60-62.

[20] Ito Title.

[21] Ito Abstract.

[22] Ito Paragraph [0143].

> bit region are acquired. Based on these address, Column (column address) is incremented and 16 parity bits are written into memory cells … .[23]

Thus, Ito appears to teach loading data read from the memory into feedback shift registers that are used to perform the logical operation to calculate check bits, and storing the calculated check bits into the memory or a parity data accumulating section. Ito also teaches that the ECC-CODEC calculates check bits for data already in memory.

Representative claim 12, by contrast, specifically recites "a means for computing an ECC of the data block; a means for shifting the data block between the means for storing and the means for controlling while computing an ECC for said data block; and a means for selectively storing the ECC in a plurality of registers while shifting the data block." Appellants submit that Eggleston, Wei and Ito do not teach or fairly suggest such a method. Ito appears to teach loading data read from the memory into feedback shift registers that are used to perform the logical operation to calculate check bits, and storing the calculated check bits into the memory or a parity data accumulating section. Ito does not appear to teach selectively storing the check bits in a plurality of registers. Ito also teaches that the ECC-CODEC calculates check bits for data already in memory. Thus, even if the teachings of Eggleston and Wei are precisely as the Office Action suggests (which Appellants do not admit), Eggleston, Wei and Ito still fail to teach or fairly suggest "a means for computing an ECC of the data block; a means for shifting the data block between the means for storing and the means for controlling **while computing an ECC for said data block**; and **a means for selectively storing the ECC in a plurality of registers while shifting the data block**."

Based on the foregoing, Appellants respectfully request that the rejection of this grouping be reversed, and the claims set for issue.

C. Section 103 Rejections over Eggleston, Wei, Ito and Acton

1. Claim 5, 11 and 17

Claims 5, 11 and 17 stands rejected as allegedly obvious over Eggleston, Wei, Ito and Acton. Claims 5, 11 and 17 are allowable for at least the same reasons as

[23] Ito Paragraph [0144].

delineated in Sections VII(B)(1). Further, claim 17 is allowable for at least the same reasons as delineated in Section VII(B)(2).

D. Conclusion

For the reasons stated above, Appellants respectfully submit that the Examiner erred in rejecting all pending claims. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to the Texas Instruments, Inc. Deposit Account No. 20-0668.

Respectfully submitted,

/Utpal D. Shah/

Utpal D. Shah
PTO Reg. No. 60,047
CONLEY ROSE, P.C.
(512) 610-3410 (Phone)
(512) 610-3456 (Fax)
AGENT FOR APPELLANTS

VIII. CLAIMS APPENDIX

1. (Previously Presented) A method, comprising:

transferring a data block between a flash memory and a memory controller;

computing an ECC for said data block while transferring the data block; and

selectively storing the ECC in a plurality of registers using a switching mechanism, the storing while transferring the data block.

2. (Original) The method of claim 1, wherein transferring the data block comprises transferring the data block between a NAND Flash memory and the memory controller.

3. (Previously Presented) The method of claim 1, wherein selectively storing the ECC further comprises

storing a first portion of the ECC in a first register; and

storing a second portion of the ECC in a second register if the first register is full.

4. (Previously Presented) The method of claim 3, wherein storing in a second register comprises selecting the second register using the switching mechanism.

5. (Original) The method of claim 1, wherein computing the ECC comprises performing the exclusive-or function.

6. (Previously Presented) A system, comprising:

a flash memory;

a controller coupled to the flash memory;

a switch coupled to the controller; and

said controller is configured to shift a data block between the flash memory and the controller, while computing an ECC for said data block; and

said system is configured to selectively store the ECC in a plurality of registers using the switch, while the controller shifts the data block.

7. (Original) The system of claim 6, wherein the flash memory is a NAND Flash memory.

8. (Previously Presented) The system of claim 6, wherein the system is configured to
store a first portion of the ECC in a first register; and
store a second portion of the ECC in an alternate register if the first register is full.

9. (Previously Presented) The system of claim 8, wherein the controller is configured to transfer contents of all registers to memory if all registers are full.

10. (Previously Presented) The system of claim 8, wherein the switch configured to select the alternate register.

11. (Previously Presented) The system of claim 6, wherein the controller is configured to compute the ECC while performing the exclusive-or function.

12. (Previously Presented) A system comprising:
a means for storing a data block;
a means for controlling the data block;
a means for computing an ECC of the data block;
a means for shifting the data block between the means for storing and the means for controlling while computing an ECC for said data block; and
a means for selectively storing the ECC in a plurality of registers while shifting the data block.

13. (Original) The system of claim 12, wherein the means for storing is a NAND Flash memory.

14. (Previously Presented) The system of claim 12, wherein the means for selectively storing the ECC is configured to

store the ECC in a first register; and

store the ECC in an alternate register if the first register is full.

15. (Previously Presented) The system of claim 12, wherein the system is configured to transfer contents of at least one register to memory if all registers are full.

16. (Previously Presented) The system of claim 14, wherein the means for selectively storing the ECC is a switch configured to select the alternate register.

17. (Previously Presented) The system of claim 12, wherein the system is configured to compute the ECC while performing the exclusive-or function.

18. (Previously Presented) A memory controller configured to couple to a memory, comprising:

a memory interface;

an ECC engine configured to compute an ECC while transferring a data block between the ECC engine and memory; and

a switching mechanism coupled to the ECC engine, the ECC engine configured to selectively store the ECC in a plurality of registers using the switching mechanism, while transferring the data block.

19. (Previously Presented) The memory controller of claim 18, further comprising:

a register bank coupled to the switching mechanism, comprising at least one register;

wherein the ECC engine configured to store the ECC in a register selected by the switching mechanism, the register having space available for ECC storage.

20. (Previously Presented) The memory controller of claim 18, wherein the controller is configured to transfer a data block while transferring the data block between the ECC engine and a flash memory.

21. (Previously Presented) The memory controller of claim 18, wherein the controller is configured to transfer a data block while transferring the data block between the ECC engine and a NAND Flash memory.

22. (Previously Presented) The memory controller of claim 18, wherein the ECC engine is configured to transfer a data block by reading the data block from memory.

23. (Previously Presented) The memory controller of claim 18, wherein the ECC engine is configured to transfer a data block by writing the data block to memory.

24. (Previously Presented) The system of claim 8, wherein the first register is in the controller.

25. (Previously Presented) The system of claim 8, wherein the alternate register is in the controller.

26. (Previously Presented) The system of claim 10, wherein the switch is in the controller.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.